

# Contents

|           |                                                                                        |           |
|-----------|----------------------------------------------------------------------------------------|-----------|
| <b>I</b>  | <b>Introduction</b>                                                                    | <b>1</b>  |
| <b>1</b>  | <b>A Continuously Available, High Transaction Capacity, Soft Real-Time DBMS Server</b> | <b>3</b>  |
| 1.1       | Motivation . . . . .                                                                   | 3         |
| 1.2       | Requirements for Continuously Available DBMS Servers . . . . .                         | 4         |
| 1.3       | Fault-Tolerance Concepts . . . . .                                                     | 5         |
| <b>2</b>  | <b>DBMS Recovery Requirements</b>                                                      | <b>9</b>  |
| 2.1       | The DBMS Service . . . . .                                                             | 9         |
| 2.2       | The DBMS Reliability . . . . .                                                         | 9         |
| 2.3       | The DBMS Failure Model . . . . .                                                       | 10        |
| 2.4       | Node and Fragment Crash Recovery, Masking and Repair . . . . .                         | 11        |
| 2.5       | The DBMS Server Architecture . . . . .                                                 | 13        |
| 2.6       | The DBMS Recovery Methods Design Goals . . . . .                                       | 14        |
| <b>II</b> | <b>Analysis of Centralised DBMS Transaction and Node Crash Recovery Approaches</b>     | <b>19</b> |
| <b>3</b>  | <b>Introduction</b>                                                                    | <b>21</b> |
| <b>4</b>  | <b>The Block Oriented Approach</b>                                                     | <b>23</b> |
| 4.1       | The Block Oriented DBMS Server Architecture . . . . .                                  | 23        |
| 4.2       | The Block Oriented Recovery Classification Scheme . . . . .                            | 25        |
| 4.3       | The Impact of Volatile DB Buffer on Transaction and Node Recoverability . . . . .      | 25        |
| 4.4       | The Log Strategy . . . . .                                                             | 26        |
| 4.5       | The DB Block Propagation Strategy . . . . .                                            | 30        |
| 4.6       | The DB Buffer Strategy . . . . .                                                       | 31        |
| 4.7       | The Checkpoint Strategy . . . . .                                                      | 33        |
| 4.8       | The Combined Block Oriented Classification Scheme . . . . .                            | 40        |
| 4.9       | Conclusion . . . . .                                                                   | 41        |
| <b>5</b>  | <b>The Record Oriented Approach</b>                                                    | <b>45</b> |
| 5.1       | The Record Oriented DBMS Server Architecture . . . . .                                 | 45        |
| 5.2       | The Record Oriented Classification Scheme . . . . .                                    | 47        |
| 5.3       | The Record Logging Strategy . . . . .                                                  | 48        |
| 5.4       | The State-Identifier Strategy . . . . .                                                | 52        |
| 5.5       | The Record Access Strategy . . . . .                                                   | 53        |
| 5.6       | The Transaction Failure Recovery Strategy . . . . .                                    | 55        |
| 5.7       | The Node Crash Recovery Strategy . . . . .                                             | 57        |
| 5.8       | The Node Crash Recovery Methods . . . . .                                              | 62        |
| 5.9       | Conclusion . . . . .                                                                   | 64        |

|            |                                                                         |            |
|------------|-------------------------------------------------------------------------|------------|
| <b>6</b>   | <b>The Compensation Oriented Approach</b>                               | <b>69</b>  |
| 6.1        | The Compensation Oriented DBMS Server Architecture                      | 69         |
| 6.2        | The Compensation Oriented Classification Scheme                         | 70         |
| 6.3        | The Record Logging Strategy                                             | 71         |
| 6.4        | The Compensation Recovery Strategy                                      | 72         |
| 6.5        | The Record Access Strategy                                              | 80         |
| 6.6        | The Transaction Failure Recovery Strategy                               | 82         |
| 6.7        | The Node Crash Recovery Strategy                                        | 84         |
| 6.8        | The Node Crash Recovery Methods                                         | 88         |
| 6.9        | Conclusion                                                              | 91         |
| <b>7</b>   | <b>The Table Oriented Approach</b>                                      | <b>97</b>  |
| 7.1        | The Table Oriented DBMS Server Architecture                             | 97         |
| 7.2        | The Table Oriented Classification Scheme                                | 99         |
| 7.3        | The Logging Strategy                                                    | 100        |
| 7.4        | The Checkpoint Strategy                                                 | 103        |
| 7.5        | The Tuple Access Strategy                                               | 105        |
| 7.6        | The Non-Sequential Recovery Strategy                                    | 111        |
| 7.7        | The Compensation Recovery Strategy                                      | 116        |
| 7.8        | The Transaction Failure Recovery Strategy                               | 116        |
| 7.9        | The Node Crash Recovery Strategy                                        | 119        |
| 7.10       | Conclusion                                                              | 123        |
| <b>III</b> | <b>The HypRa Recovery Approach</b>                                      | <b>131</b> |
| <b>8</b>   | <b>Introduction</b>                                                     | <b>133</b> |
| 8.1        | The HypRa DBMS Server Architecture                                      | 134        |
| <b>9</b>   | <b>The Hardware and Basic Software Design</b>                           | <b>137</b> |
| 9.1        | The Hardware Architecture                                               | 137        |
| 9.2        | The External Communication                                              | 138        |
| 9.3        | Replaceable Hardware Units                                              | 140        |
| 9.4        | Depends-On Relations Between Hardware Servers                           | 140        |
| 9.5        | Disaster-Tolerant Hardware Design                                       | 142        |
| 9.6        | The Communication Server                                                | 142        |
| <b>10</b>  | <b>The HypRa DBMS Server Software Design</b>                            | <b>147</b> |
| 10.1       | The Distributed HypRa DBMS Architecture                                 | 147        |
| 10.2       | The Distributed Transaction Model                                       | 147        |
| 10.3       | Concurrency Control in the HypRa Server                                 | 149        |
| 10.4       | The Nested Transaction Model                                            | 149        |
| 10.5       | Buffer Management                                                       | 150        |
| 10.6       | Data Distribution                                                       | 151        |
| 10.7       | Dynamic Reconfiguration of Data as a Mechanism for Repairing Faults     | 153        |
| 10.8       | Gradual Reduction of Fragment Availability                              | 154        |
| <b>11</b>  | <b>The Location and Replication Independent Tuple Recovery Strategy</b> | <b>159</b> |
| 11.1       | The Tuple State-Identifier Policy                                       | 160        |
| 11.2       | The Primary and Hot Stand-By Tuple Replica Synchronisation Policy       | 162        |
| 11.3       | The Primary Key Update Policies                                         | 164        |
| 11.4       | The Multi-Level State-Identifier Policy                                 | 165        |
| 11.5       | Conclusion                                                              | 166        |

|                                                                                         |            |
|-----------------------------------------------------------------------------------------|------------|
| <b>12 The Log Distribution Strategy</b>                                                 | <b>169</b> |
| 12.1 The Log Distribution Policy . . . . .                                              | 169        |
| 12.2 The Tuple Log Record Identification Policies . . . . .                             | 172        |
| 12.3 The Tuple Log Record Identification Policy Used at Hot Stand-By Replicas . . . . . | 173        |
| 12.4 The Log Subfragmentation Policy . . . . .                                          | 175        |
| 12.5 The Fragment Concatenation Policy . . . . .                                        | 176        |
| 12.6 The Node Internal Log . . . . .                                                    | 177        |
| 12.7 The Tuple Identification Concept Policies . . . . .                                | 178        |
| 12.8 Conclusion . . . . .                                                               | 179        |
| <b>13 The Neighbour Write-Ahead Logging Strategy</b>                                    | <b>181</b> |
| 13.1 The Stable Storage Policies . . . . .                                              | 182        |
| 13.2 The Neighbour Write Ahead Log Policy . . . . .                                     | 183        |
| 13.3 The Neighbour Write-Ahead Log Garbage Collection Rules . . . . .                   | 184        |
| 13.4 The Update Channel Protocol . . . . .                                              | 186        |
| 13.5 The Deferred Execution Policy . . . . .                                            | 187        |
| 13.6 The nWAL Distributed Transaction Model . . . . .                                   | 190        |
| 13.7 The nWAL Transaction Model . . . . .                                               | 191        |
| 13.8 The nWAL Checkpointing Policy . . . . .                                            | 193        |
| 13.9 The Main Memory Log Overflow Policy . . . . .                                      | 195        |
| 13.10 Dynamic Replication to Preserve DB Consistency . . . . .                          | 196        |
| 13.11 Block Split and Merge . . . . .                                                   | 198        |
| 13.12 The Global Undo Space Reservation Policy . . . . .                                | 198        |
| 13.13 Conclusion . . . . .                                                              | 199        |
| <b>14 The Recovery Strategies</b>                                                       | <b>205</b> |
| 14.1 The Take-Over Strategy . . . . .                                                   | 205        |
| 14.2 The Transaction Failure Recovery Strategy . . . . .                                | 208        |
| 14.3 The Fast Node Crash Recovery Strategy . . . . .                                    | 212        |
| 14.4 Conclusion . . . . .                                                               | 225        |
| <b>IV Other High Availability DBMS Approaches</b>                                       | <b>229</b> |
| <b>15 High Availability Approaches</b>                                                  | <b>231</b> |
| 15.1 The Symmetric Synchronisation Strategy . . . . .                                   | 231        |
| 15.2 The Asymmetric Synchronisation Strategy . . . . .                                  | 233        |
| 15.3 The 1-Safe Consistency Policy . . . . .                                            | 236        |
| 15.4 Evaluation Criteria . . . . .                                                      | 238        |
| <b>16 Shared Disc Approaches</b>                                                        | <b>239</b> |
| 16.1 The Tandem Non-Stop System . . . . .                                               | 239        |
| 16.2 The IBM Extended Recovery Facility . . . . .                                       | 241        |
| <b>17 Shared-Nothing Approaches</b>                                                     | <b>245</b> |
| 17.1 The Tandem Remote Duplicate Database Facility . . . . .                            | 245        |
| 17.2 The E-Net Remote Recovery Data and Log Apply Facility . . . . .                    | 246        |
| 17.3 Oracle7 Symmetric Replication Facility . . . . .                                   | 248        |
| 17.4 The Sybase Replication Server . . . . .                                            | 250        |
| 17.5 The CA-OpenIngres/Replicator . . . . .                                             | 251        |
| 17.6 The INFORMIX-OnLine Dynamic Server . . . . .                                       | 252        |
| 17.7 The IBM Remote Site Recovery . . . . .                                             | 253        |
| 17.8 Telecom Switch Internal DBMSs . . . . .                                            | 255        |
| 17.9 Academic Research Work . . . . .                                                   | 255        |

|                                                                            |                |
|----------------------------------------------------------------------------|----------------|
| <b>V Conclusion and Future Work</b>                                        | <b>259</b>     |
| <b>18 Conclusion</b>                                                       | <b>261</b>     |
| 18.1 Lessons Learned from Centralised DBMS Recovery Approaches . . . . .   | 261            |
| 18.2 Main Contributions from the HypRa Recovery Approach . . . . .         | 262            |
| 18.3 Maintaining a Continuously Available Transaction Server . . . . .     | 263            |
| <b>19 Future Work</b>                                                      | <b>265</b>     |
| 19.1 Follow Up Threads . . . . .                                           | 265            |
| 19.2 Commercialising HypRa . . . . .                                       | 266            |
| <br><b>Bibliography</b>                                                    | <br><b>267</b> |
| <br><b>Index</b>                                                           | <br><b>281</b> |
| <br><b>VI Appendices</b>                                                   | <br><b>291</b> |
| <b>A Basic Concepts</b>                                                    | <b>293</b>     |
| A.1 Datamodel, Database, and Database Management System Concepts . . . . . | 293            |
| A.2 Distributed Database Concepts . . . . .                                | 293            |
| A.3 The DBMS Mapping Architecture . . . . .                                | 294            |
| A.4 The Table Datamodel . . . . .                                          | 296            |
| A.5 Table Datamodel Operations . . . . .                                   | 297            |
| A.6 The Record Datamodel . . . . .                                         | 298            |
| A.7 The Block Datamodel . . . . .                                          | 299            |
| A.8 The Transaction Model . . . . .                                        | 299            |
| A.9 Concurrency Control Concepts . . . . .                                 | 300            |